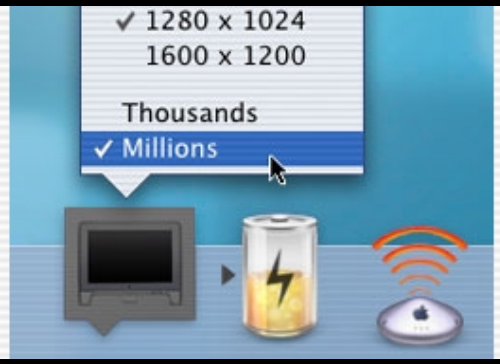
Session 109

Graphics for Games



David Hill
Printing, Graphics, and Games DTS
Apple Worldwide Developer Relations

Introduction

- Mac OS X solutions
  - Display management
  - Mixing core graphics and QuickDraw
  - QuickDraw tidbits
  - Tear-free blitting
  - OpenGL changes
- WWDC games roadmap

Introduction

- What won't be covered in detail
  - QuickDraw
  - QuickTime
  - General core graphics introduction
  - OpenGL beyond basic Mac OS X setup

Display Management

DrawSprocket

- Primarily for ease of porting
- Callable from CFM and Mach-O
- Draw directly into front buffer
- No DSpUserSelectContext

Display Management

CGDirectDisplay

- Enumerating displays
- Enumerating available video modes
- Capturing displays
- Switching video modes
- Accessing screen memory directly

CGDirectDisplay

Enumerating displays

```
CGDirectDisplayID displays[ kMaxDisplays ];
CGDisplayErr err = CGGetActiveDisplayList(
kMaxDisplays, displays, &actualNumberOfDisplays );
if ( err == kCGDisplayNoErr )
{
    for ( int i = 0; i < actualNumberOfDisplays; i++ )
    {
        //  Do something with displays[ i ];
    }
}
```

CGDirectDisplay

Enumerating displays

```
CGDirectDisplayID displays[ kMaxDisplays ];
CGDisplayErr err = CGGetActiveDisplayList(
kMaxDisplays, displays, &actualNumberOfDisplays );
if ( err == kCGDisplayNoErr )
{
    for ( int i = 0; i < actualNumberOfDisplays; i++ )
    {
        //  Do something with displays[ i ];
    }
}
```

CGDirectDisplay

Enumerating displays

```
CGDirectDisplayID displays[ kMaxDisplays ];
CGDisplayErr err = CGGetActiveDisplayList(
kMaxDisplays, displays, &actualNumberOfDisplays );
if ( err == kCGDisplayNoErr )
{
    for ( int i = 0; i < actualNumberOfDisplays; i++ )
    {
        //  Do something with displays[ i ];
    }
}
```

CGDirectDisplay

Investigating a display

```
size_t sizeX = CGDisplayPixelsWide(displays[ i ]);
size_t sizeY = CGDisplayPixelsHigh(displays[ i ]);
size_t rowBytes = CGDisplayBytesPerRow(displays[ i ]);

//  Only valid for a captured display
UInt8* baseAddr = CGDisplayBaseAddress(displays[ i ]);
```

CGDirectDisplay

Investigating a display

```
size_t sizeX = CGDisplayPixelsWide(displays[ i ]);
size_t sizeY = CGDisplayPixelsHigh(displays[ i ]);
size_t rowBytes = CGDisplayBytesPerRow(displays[ i ]);

//  Only valid for a captured display
UInt8* baseAddr = CGDisplayBaseAddress(displays[ i ]);
```

CGDirectDisplay

Investigating a display

```
size_t sizeX = CGDisplayPixelsWide(displays[ i ]);
size_t sizeY = CGDisplayPixelsHigh(displays[ i ]);
size_t rowBytes = CGDisplayBytesPerRow(displays[ i ]);

//  Only valid for a captured display
UInt8* baseAddr = CGDisplayBaseAddress(displays[ i ]);
```

CGDirectDisplay

Enumerating available video modes

```
CFArrayRef modes =
    CGDisplayAvailableModes( displays[ i ] );
if (modes != NULL ) {
    CFIndex numOfModes = CFArrayGetCount( modes );
    for ( int j = 0; j < numOfModes ; j++ ) {
        CFDictionaryRef mode =
                    CFArrayGetValueAtIndex( modes, j );
        //  Do something with the mode dictionary
    }
}
```

CGDirectDisplay

Enumerating available video modes

```
CFArrayRef modes =
    CGDisplayAvailableModes( displays[ i ] );
if (modes != NULL ) {
    CFIndex numOfModes = CFArrayGetCount( modes );
    for ( int j = 0; j < numOfModes ; j++ ) {
        CFDictionaryRef mode =
                    CFArrayGetValueAtIndex( modes, j );
        //  Do something with the mode dictionary
    }
}
```

CGDirectDisplay

Enumerating available video modes

```
CFArrayRef modes =
    CGDisplayAvailableModes( displays[ i ] );
if (modes != NULL ) {
    CFIndex numOfModes = CFArrayGetCount( modes );
    for ( int j = 0; j < numOfModes ; j++ ) {
        CFDictionaryRef mode =
                CFArrayGetValueAtIndex( modes, j );
        //  Do something with the mode dictionary
    }
}
```

CGDirectDisplay

Investigating a mode

```
if ( CFDictionaryGetValue(mode,
        kCGDisplayModeUsableForDesktopGUI) ==
        kCFBooleanTrue )
   msg = "Supports Aqua GUI";

CFNumberRef value;
int num = 0;
if ( (value = CFDictionaryGetValue(mode, key)) != NULL )
   CFNumberGetValue(value, kCFNumberIntType, &num);
```

CGDirectDisplay

Investigating a mode

```
if ( CFDictionaryGetValue(mode,
        kCGDisplayModeUsableForDesktopGUI) ==
        kCFBooleanTrue )
   msg = "Supports Aqua GUI";

CFNumberRef value;
int num = 0;
if ( (value = CFDictionaryGetValue(mode, key)) != NULL )
   CFNumberGetValue(value, kCFNumberIntType, &num);
```

CGDirectDisplay

Investigating a mode

```
if ( CFDictionaryGetValue(mode,
        kCGDisplayModeUsableForDesktopGUI) ==
        kCFBooleanTrue )
   msg = "Supports Aqua GUI";

CFNumberRef value;
int num = 0;
if ( (value = CFDictionaryGetValue(mode, key)) != NULL )
   CFNumberGetValue(value, kCFNumberIntType, &num);
```

CGDirectDisplay

Investigating a mode

```
if ( CFDictionaryGetValue(mode,
        kCGDisplayModeUsableForDesktopGUI) ==
        kCFBooleanTrue )
   msg = "Supports Aqua GUI";

CFNumberRef value;
int num = 0;
if ( (value = CFDictionaryGetValue(mode, key)) != NULL )
   CFNumberGetValue(value, kCFNumberIntType, &num);
```

CGDirectDisplay

Switching modes

```
CFDictionaryRef originalMode = CGDisplayCurrentMode( bestDisplayID );
CGDisplayCapture( bestDisplayID );
CGDisplaySwitchToMode( bestDisplayID, bestMode );
CGDisplayHideCursor( bestDisplayID );

//  Your game code here
//  Use CreateNewPortForCGDisplayID to get a CGrafPtr
//  Don’t forget to call DisposePort() when you’re done
```

CGDirectDisplay

Switching modes

```
CFDictionaryRef originalMode = CGDisplayCurrentMode( bestDisplayID );
CGDisplayCapture( bestDisplayID );
CGDisplaySwitchToMode( bestDisplayID, bestMode );
CGDisplayHideCursor( bestDisplayID );

//  Your game code here
//  Use CreateNewPortForCGDisplayID to get a CGrafPtr
//  Don’t forget to call DisposePort() when you’re done
```

CGDirectDisplay

Switching modes

```
CFDictionaryRef originalMode = CGDisplayCurrentMode( bestDisplayID );
CGDisplayCapture( bestDisplayID );
CGDisplaySwitchToMode( bestDisplayID, bestMode );
CGDisplayHideCursor( bestDisplayID );

//  Your game code here
//  Use CreateNewPortForCGDisplayID to get a CGrafPtr
//  Don’t forget to call DisposePort() when you’re done
```

CGDirectDisplay

Switching modes

```
CFDictionaryRef originalMode = CGDisplayCurrentMode( bestDisplayID );
CGDisplayCapture( bestDisplayID );
CGDisplaySwitchToMode( bestDisplayID, bestMode );
CGDisplayHideCursor( bestDisplayID );

//  Your game code here
//  Use CreateNewPortForCGDisplayID to get a CGrafPtr
//  Don’t forget to call DisposePort() when you’re done
```

CGDirectDisplay

Switching modes

```
CFDictionaryRef originalMode = CGDisplayCurrentMode( bestDisplayID );
CGDisplayCapture( bestDisplayID );
CGDisplaySwitchToMode( bestDisplayID, bestMode );
CGDisplayHideCursor( bestDisplayID );

//  Your game code here
//  Use CreateNewPortForCGDisplayID to get a CGrafPtr
//  Don’t forget to call DisposePort() when you’re done
```

CGDirectDisplay

Shutting down

```
//  Your game code here

CGDisplayShowCursor( bestDisplayID );
CGDisplaySwitchToMode( bestDisplayID, originalMode );
CGDisplayRelease( bestDisplayID );
```

CGDirectDisplay

Shutting down

```
//  Your game code here

CGDisplayShowCursor( bestDisplayID );
CGDisplaySwitchToMode( bestDisplayID, originalMode );
CGDisplayRelease( bestDisplayID );
```

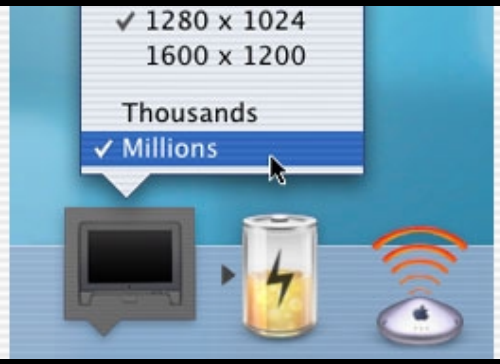
CGDirectDisplay

Shutting down

```
//  Your game code here

CGDisplayShowCursor( bestDisplayID );
CGDisplaySwitchToMode( bestDisplayID, originalMode );
CGDisplayRelease( bestDisplayID );
```

Demo



Display Management

CGDirectPalette and CGDirectDisplay

- Creating Palettes
- Manipulating Palettes
- Using Palettes
- Manipulating Gamma Tables

CGDirectPalette

Creating palettes

- CGPaletteCreateDefaultColorPalette
- CGPaletteCreateWithDisplay
- CGPaletteCreateWithCapacity
- CGPaletteCreateWithSamples
- CGPaletteCreateWithByteSamples
- CGPaletteRelease

CGDirectPalette

Manipulating palettes

- CGPaletteGetColorAtIndex
- CGPaletteGetIndexForColor
- CGPaletteSetColorAtIndex
- CGPaletteIsEqualToPalette
- CGPaletteCreateFromPaletteBlendedWithColor

CGDirectDisplay

Using palettes

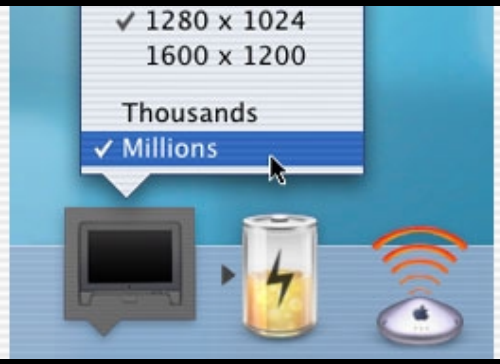
- CGDisplayCanSetPalette
- CGDisplaySetPalette

CGDirectDisplay

Manipulating gamma tables

- CGSetDisplayTransferByFormula
- CGGetDisplayTransferByFormula
- CGSetDisplayTransferByTable
- CGGetDisplayTransferByTable
- CGSetDisplayTransferByByteTable

Demo



Graphics for Games

Mixing QuickDraw and Core Graphics

- Accessing CG from a QD port
- Coordinate systems
- Pixel addressing
- Clipping

Accessing CG From a QD Port

Creating a CG Context

```
CGContextRef contextRef;
OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

//  Do your CG drawing here

CGContextRelease( contextRef );
```

Accessing CG From a QD Port

Creating a CG Context

```
CGContextRef contextRef;
OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

//  Do your CG drawing here

CGContextRelease( contextRef );
```

Core Graphics

Coordinate systems

- How is it different from QuickDraw?
  - Origin starts in the bottom left
  - Positive Y axis points up
  - No concept of bounds

Core Graphics

Coordinate systems

- What can we do about it?
  - Fix coordinates
    - CGSyncContextOriginWithPort
    - CGContextTranslateCTM
    - CGContextScaleCTM
  - Remember bounds from port or view

Coordinate Systems

Making Core Graphics match QuickDraw

```
CGContextRef contextRef;
OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

CGContextSyncOriginWithPort( contextRef, portPtr );
CGContextTranslateCTM( contextRef, 0, portHeight );
CGContextScaleCTM( contextRef, 1, -1 );

//  CG coordinates match QD's
//  However, text draws upside down!
```

Coordinate Systems

Making Core Graphics match QuickDraw

```
CGContextRef contextRef;
OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

CGContextSyncOriginWithPort( contextRef, portPtr );
CGContextTranslateCTM( contextRef, 0, portHeight );
CGContextScaleCTM( contextRef, 1, -1 );

//  CG coordinates match QD's
//  However, text draws upside down!
```

Coordinate Systems

Making Core Graphics match QuickDraw

```
CGContextRef contextRef;
OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

CGContextSyncOriginWithPort( contextRef, portPtr );
CGContextTranslateCTM( contextRef, 0, portHeight );
CGContextScaleCTM( contextRef, 1, -1 );

//  CG coordinates match QD's
//  However, text draws upside down!
```

Coordinate Systems

Making Core Graphics match QuickDraw

```
CGContextRef contextRef;
OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

CGContextSyncOriginWithPort( contextRef, portPtr );
CGContextTranslateCTM( contextRef, 0, portHeight );
CGContextScaleCTM( contextRef, 1, -1 );

//  CG coordinates match QD's
//  However, text draws upside down!
```

Coordinate Systems

Making Core Graphics match QuickDraw

```
CGContextRef contextRef;
OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

CGContextSyncOriginWithPort( contextRef, portPtr );
CGContextTranslateCTM( contextRef, 0, portHeight );
CGContextScaleCTM( contextRef, 1, -1 );

//  CG coordinates match QD's
//  However, text draws upside down!
```

Core Graphics

Current Transformation Matrix (CTM)

- Translate
- Scale
- Rotate
- Skew (arbitrary affine transform)

Coordinate Systems

Transforming primitives

```
CGContextRef contextRef;
OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

CGContextScaleCTM( contextRef, scaleX, scaleY );
CGContextTranslateCTM( contextRef, transX, transY );
CGContextRotateCTM( contextRef, angleInRadians );
CGContextConcatCTM( contextRef, affineTransform );
```

Coordinate Systems

Transforming primitives

```
CGContextRef contextRef;
OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

CGContextScaleCTM( contextRef, scaleX, scaleY );
CGContextTranslateCTM( contextRef, transX, transY );
CGContextRotateCTM( contextRef, angleInRadians );
CGContextConcatCTM( contextRef, affineTransform );
```

Coordinate Systems

Transforming primitives

```
CGContextRef contextRef;
OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

CGContextScaleCTM( contextRef, scaleX, scaleY );
CGContextTranslateCTM( contextRef, transX, transY );
CGContextRotateCTM( contextRef, angleInRadians );
CGContextConcatCTM( contextRef, affineTransform );
```

Coordinate Systems

Transforming primitives

```
CGContextRef contextRef;
OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

CGContextScaleCTM( contextRef, scaleX, scaleY );
CGContextTranslateCTM( contextRef, transX, transY );
CGContextRotateCTM( contextRef, angleInRadians );
CGContextConcatCTM( contextRef, affineTransform );
```

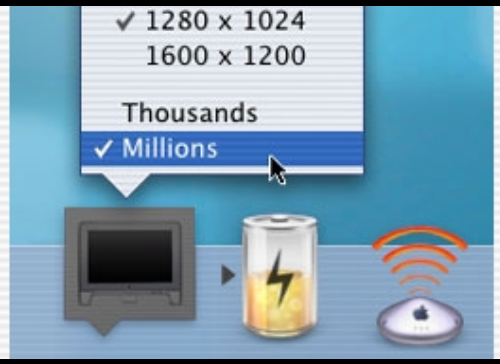
Coordinate Systems

Transforming primitives

```
CGContextRef contextRef;
OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

CGContextScaleCTM( contextRef, scaleX, scaleY );
CGContextTranslateCTM( contextRef, transX, transY );
CGContextRotateCTM( contextRef, angleInRadians );
CGContextConcatCTM( contextRef, affineTransform );
```

Demo



Core Graphics

Pixel addressing

- CGContexts don't normally allow direct access
- CGBitmapContextCreate
  - Data pointer or NULL
  - Width and height
  - Bits per component
  - Color space

Pixel Addressing

Rendering to an offscreen bitmap

```
CGContextRef contextRef;
contextRef = CGBitmapContextCreate ( dataPtr, width,
                                     height, bitsPerComponent,
                                     colorSpace, alphaInfo );

//  Proceed with normal Core Graphics operations
//  Or access the bits in dataPtr directly
```

Pixel Addressing

Rendering to an offscreen bitmap

```
CGContextRef contextRef;
contextRef = CGBitmapContextCreate ( dataPtr, width,
                                     height, bitsPerComponent,
                                     colorSpace, alphaInfo );

//  Proceed with normal Core Graphics operations
//  Or access the bits in dataPtr directly
```

Core Graphics

Clipping

- CGContext created for port doesn't inherit clip region from CGrafPort
- What can we do?
  - Grab clip region from CGrafPort
  - Call ClipCGContextToRegion

Clipping

Clipping a context to the QuickDraw clip

```
CGContextRef contextRef;          OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

//  Make the CG clip match that of the QD port
RgnHandle clipRgn = NewRgn();
Rect portRect;   GetPortBounds( portPtr, &portRect );
clipRgn = GetPortClipRegion( portPtr, clipRgn );
ClipCGContextToRegion( contextRef,
                &portRect,clipRgn );
DisposeRgn( clipRgn );
```

Clipping

Clipping a context to the QuickDraw clip

```
CGContextRef contextRef;          OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

//  Make the CG clip match that of the QD port
RgnHandle clipRgn = NewRgn();
Rect portRect;   GetPortBounds( portPtr, &portRect );
clipRgn = GetPortClipRegion( portPtr, clipRgn );
ClipCGContextToRegion( contextRef,
                &portRect,clipRgn );
DisposeRgn( clipRgn );
```

Clipping

Clipping a context to the QuickDraw clip

```
CGContextRef contextRef;          OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

//  Make the CG clip match that of the QD port
RgnHandle clipRgn = NewRgn();
Rect portRect;   GetPortBounds( portPtr, &portRect );
clipRgn = GetPortClipRegion( portPtr, clipRgn );
ClipCGContextToRegion( contextRef,
                &portRect,clipRgn );
DisposeRgn( clipRgn );
```

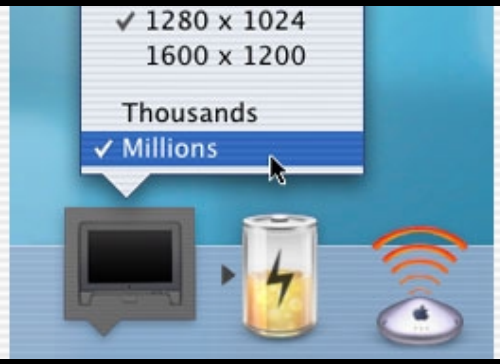
Clipping

Clipping a context to the QuickDraw clip

```
CGContextRef contextRef;          OSStatus err;
err = CreateCGContextForPort( portPtr, &contextRef );

//  Make the CG clip match that of the QD port
RgnHandle clipRgn = NewRgn();
Rect portRect;   GetPortBounds( portPtr, &portRect );
clipRgn = GetPortClipRegion( portPtr, clipRgn );
ClipCGContextToRegion( contextRef,
                &portRect,clipRgn );
DisposeRgn( clipRgn );
```

Demo



Graphics for Games

QuickDraw tidbits

- CreateNewPort
  - May not always point to the screen
  - Creates CGrafPorts, not GrafPorts
- QDFlushPortBuffer
  - Animation
  - VBL syncing

Tear-Free Blitting

Things you need to know

- OpenGL can sync to the VBL

```
aglSetInteger(aglContext,AGL_SWAP_INTERVAL,1);
```

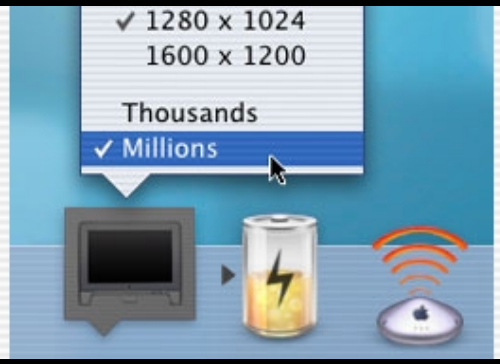
- Windows are double buffered
  - Use window's back buffer instead of your own
  - Window buffer flushes sync to the VBL where supported by hardware

Tear-Free Blitting

Things you need to know

- Mac OS X is a fully pre-emptive OS
- Direct screen access is hard to sync to the VBL
- Let us do the VBL syncing for you

Demo



OpenGL Changes

OpenGL on Mac OS X

- Important differences
- Integration with DrawSprocket

OpenGL Changes

Important differences

- aglFullScreen
- New features
  - Stencil Buffers
  - Aux Buffers
- New extensions
- Supported in Carbon and Cocoa

OpenGL Changes

More differences

- aglUpdateContext is more critical
- aglSetDrawable
  - Don’t pass in WindowRef cast to CGrafPtr
  - Use GetWindowPort

OpenGL Changes

Integration with DrawSprocket

- Mac OS 9
  - Window on top of DSp
  - Multiple monitors and hardware acceleration
- Mac OS X
  - Call DSpContext_GetFrontBuffer
  - Pass DSp front buffer CGrafPtr to aglSetDrawable

Graphics for Games

Summary

- Use DrawSprocket or CGDirectDisplay
- You can mix Core Graphics and QuickDraw
- Mac OS X double buffers so you don't have to
- Watch out for minor QuickDraw changes
- OpenGL on Mac OS X

Roadmap

404 OpenGL: High Performance 2D	Room J2 **Wed., 3:30pm**
405 OpenGL: Geometry & Modeling	Room J2 **Wed., 5:00pm**
408 OpenGL: Optimization	Room J2 **Thur., 9:00am**
409 OpenGL: Advanced Rendering	Room J2 **Thur., 10:30am**
FF110 Feedback Forum: OpenGL	Room J1 **Thur., 5:00pm**

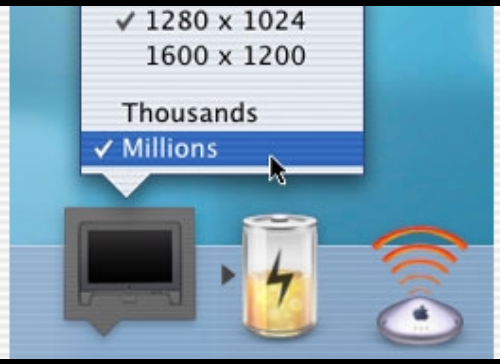
Roadmap (Cont.)

135 Input for Games	Room J2 **Fri., 10:30am**
136 Sound & Networking for Games	Room J2 **Fri., 2:00pm**
FF120 Feedback Forum: Games Technologies	Room J1 **Fri., 5:00pm**

Who to Contact

David Hill
Printing, Graphics, and Games DTS
Apple Worldwide Developer Relations
dts@apple.com

Q&A



Geoff Stahl
3D Graphics & Games DTS Engineer
gstahl@apple.com

http://developer.apple.com/events/urls.html

WWDC

Worldwide Developers Conference 2001